**AMENDMENTS TO THE SPECIFICATION:**

Kindly replace the paragraph bridging pages 9 and 10, with the following amended paragraph:

Reference numerals 1 to 15 denote general-purpose registers R0 to R15 for storing data or addresses. The general-purpose R13 is a link (LINK) register for storing a return address for a subroutine jump. The general-purpose register R15 is a register for a stack pointer (SP) including an interruption stack pointer (SPI) 16 and a user stack pointer (SPU) 17. The general-purpose register R15 is switched, by a processor status word (PSW) described later, between the interruption stack pointer (SPI) 16 and the user stack pointer (SPU) 17. The SPI 16 and the SPU 17 will hereinafter be generically referred to as a stack pointer (SP). The number of each of the general-purpose registers is specified in a 4-bit register specification field unless otherwise specified. The data processor according to this embodiment includes an instruction for specifying a pair of registers, for example, R0-1 and R1-2. In this instance, the pair of registers are specified in such a manner [[hat]] that an even-numbered register is specified to thereby implicitly specify the corresponding register of the odd number that equals the even number plus one.

Kindly replace the paragraph beginning at page 10, line 9, with the following amended paragraph:

Reference numerals 21 to [[3]] 30 denote 16-bit control registers CR0 to CR3 and CR6 to CR11. The number of each control register is also specified in a 4-bit register specification field as is the case with the general-purpose registers. The

control register CR0-21 is one that stores a processor status word (PSW) including a bit for specifying the operation mode and a flag indicating the result of operation.

Kindly replace the paragraph bridging pages 35 and 36 with the following amended paragraph:

In the E stage 403, ~~Updating~~ updating of the PC value independent of the instruction to be executed and repeat control are also performed. Upon each start of processing a 32-bit instruction, the PC unit 118 transfers the held in the latch 192 to the EPC 194 under the control of the control unit 112. The NPC 119 holds the address of the instruction to be processed next. When a jump occurs in the E stage 403, the [[ump]] jump address generated by the ALU 153 of the first operation unit 116 is transferred via the JA bus to the NPC 191 to initialize it. For sequential processing of instructions, upon each start of processing of the 32-bit instructions, the PC unit 118 writes back to the NPC unit 191 the 1-incremented value by the incrementor 193 under the control of the control unit 112. At the start of processing of the repeat block last instruction, the NPC 191 receives the first address of the repeat block from the latch 185. At the start of processing of the repeat processing last instruction, the NPC 191 receives from the latch 174 the address of the next instruction. And, in the clock cycle in which to terminate the processing of the repeat block last instruction, the PC unit 118 transfers from the latch 176 to the RPT_C register 188 the count value already decremented by 1 in the IF stage 401 under the control of the control unit 112. In the clock cycle in which to terminate the processing of the repeat processing last instruction, the control unit 112 clears the RP bits 43 of the PSW latch 222 to zero.

Kindly replace the paragraph beginning at page 37, line 24, with the following amended paragraph:

Next, a description will be given of an example of processing of sub-instructions.

Kindly replace the paragraph bridging pages 41 and 42 with the following amended paragraph:

In the case of executing an AND3 instruction in I1, the data held in the general-purpose register R10-11 and immediate value data "3" are ORed for each bit, and the result of the ORing are written in the general-purpose register R11-12. As shown in I5 in Fig. 16, the value held in the general-purpose register R11 is specified by the register number Rsrc1 in the repeat instruction. The SRLI instruction in I2b is an arithmetic 2-bit right [[shit]] shift instruction for the value held in the general-purpose register R10-11. The value held in the general-purpose register R10-11 is specified by the register number Rsrc2 in the repeat instruction. For example, when N in Equation (1) is 10, the initial value set in the general-purpose register R10-11 is "9", and as the result of the execution of the instructions I1 and I2, "2" and "1" are written in the general-purpose registers R10-11 and R11-12, respectively. The NOP instruction in I3b is a no operation instruction. This instruction is provided for the alignment of the instruction code, and even if it is executed, no effective operation is performed. The CLRAC instruction in I4b is an instruction for clearing the accumulator A0-31 to zero.

Kindly replace the paragraph bridging pages 42 and 43 with the following amended paragraph:

In the first clock cycle, the PC unit 118 outputs the PC value of the REP instruction onto the S3 bus 303 from the EPC 194 under the control of the control unit 184. The AA latch 151 in the first operation unit 116 reads therein the PC value output on the S3 bus 303. The AB latch 152 reads therein the displacement value that is specified by disp16 of the REP instruction provided from the first decoder 113. The ALU 153 in the first operation unit 116 adds together the values held in the AA and AB latches 151 and 152 to calculate the address of the instruction I9 that is the address of the last instruction in the repeat block. The first operation unit 116 provides the result of the calculation by the ALU 153 onto the JA bus 323 under the control of the control unit 112, and the RPR_E register 186 in the PC unit 118 reads therein the result of the calculation provided on the JA bus 323. Thereafter, the incrementor 173 in the PC unit 118 increments the value in the RPT_E register 186 by 1. The incremented value is written in the latch 174. As a result, the latch 174 holds the address of the next instruction in the repeat block, that is, the address of the instruction I10. In parallel to the above processing the value, which is specified by the register number Rsrc2 in the REP instruction, that is, [[he]] the value held in the general-purpose register R10-11, is transferred via the S1 bus 301 to the latch 169 in the first operation unit 116, and further, it is written in both of the RPT_C register 188 and the TRPT_C register 175 via the selector 155 and the D1 bus 311. And the value thus written in the RPT_C register 188 is further transferred to the latch 179.

Kindly replace the paragraph beginning at page 48, line 12, with the following amended paragraph:

When a jump instruction is executed in the E stage 403 during repeat processing, the jump processing takes precedence over the repeat-associated preprocessing. In this instance, the ALU 153 of the first operation unit 116 outputs its generated jump address to the JA bus 323, and the NPC 191 and the IA register 181 in the PC unit 118 read therein the jump address on the JA bus 323. When a jump of the execution stage is caused, the PC unit 118 transfers the value held in the latch 179 to the TRPT_C register 175, clearing the preceding update information held therein. However, when the jump is caused by the last instruction in [[he]] the repeat block, however, the PC unit 118 does not transfer the value held in the latch 179 to the TRPT_C register 175; the reason for this is to reflect the updated information in the RPT_C register 188.